Remarks:

In response to the rejection of claims 46-62 under the doctrine of obviousness-type double patenting, a terminal disclaimer is filed herewith.

Claims 46-62 stand rejected under 35 U.S.C. 112, second paragraph, as being indefinite. In response, claims 46 and 55 are hereby cancelled and claims 47, 50, 51, and 53, 54, 56, and 59 are hereby amended. Claims 47, 50, 56, and 59 are rewritten in independent form, with the text "whether to accept" replacing the text "whether accept" from claims 46 and 55 (in response to the requirement for clarification in paragraph 10 of the Office Action). Applicant contends for the following reasons that claims 47-54 and 56-62 as amended satisfy the requirements of 35 U.S.C. 112.

Claims 47, 50, 51, 53, 54, 56, and 59 are amended in response to specific comments and requirements in the Office Action's discussion of the rejection under 35 U.S.C. 112, second paragraph.

Claims 47-54 and 56-62 also stand rejected under 35 U.S.C. 112, second paragraph, as being indefinite on the basis that it is unclear what is the relation between the recited terms "frame data," "packet data," and "packet(s)." Paragraph 13 of the Office Action states that:

> frame (frame data) are defined at a "lower" "layer" than packets (packet data). This means, frames are the constituent parts of packets. Packets are not the constituent parts of frames as currently claimed.

Applicant disagrees with the Office Action's characterization of the terms "packets," "frames," "packet data," and "frame data." Instead, Applicant respectfully contends that these terms are used unambiguously in the specification and claims in a different sense that is unambiguous and consistent with conventional usage of these terms. In the specification and claims, the terms "frame" and "packet" are used interchangeably, and "frame data" and "packet data" are used interchangeably. This is appropriate because these terms are used in the context of describing operations at the datalink layer of the ISO OSI networking model (sometimes referred to as the protocol stack). The datalink layer of the protocol stack is the layer immediately above the physical layer. It is believed clear from the specification, which

shows an embodiment of the invention (system management controller 110 of Fig. 1) coupled to physical layer 108 (of Fig. 1), that the recited media access control (e.g., MAC 124 of Fig. 1) performs filtering (on incoming data) at the datalink layer of the protocol stack.

Units of data at the datalink layer of the protocol stack are conventionally referred to interchangeably as "frames" or "packets," as they are in the specification. For example, the attached three-page document entitled "Ethernet Tutorial" (which is part of the Brighton University Resource Kit for Students, by John English, Sixth Edition, August 2001) downloaded from the Internet (from http://burks.bton.ac.uk/burks/pcinfo/hardware/ethernet/framepac.htm) on May 24, 2005, explains conventional synonymous usage of these terms with reference to units of data at the datalink layer of the protocol stack. Similarly, data at the datalink layer of a protocol stack are frequently referred to interchangeably as "frame data" and "packet data" in the art, as they are in the specification and claims. In the claims, the recited "incoming frame data" comprise packets. Although the claims do not recite that "incoming frame data" comprise "frames," it is intended that the term "frames" could be substituted for the term "packets" in the claims without changing the scope of the claims.

The claims as amended do not require (either expressly or implicitly) that an incoming "frame" comprise "packets" as indicated in Paragraph 13 of the Office Action. Rather, each occurrence of the word "frame" in the claims is as part of the phrase "frame data." For example, claims 47 and 50 unambiguously refer to "each packet of the incoming frame data." This phrase in claims 47 and 50 implies (consistent with the specification) that the incoming frame data comprise packets, and refers to each individual one of these packets.

Applicant disagrees with the assertion in Paragraph 20 of the Office Action that "the subject matter as set forth in the claims was well known at the time of the invention" and respectfully requests that the Examiner identify any reference of record (other than cited U.S. Patent 6,377,998) over which the Examiner contends that the claims, as amended, are unpatentable.

Claims 46-62 stand rejected under 35 U.S.C. 102(e) as being anticipated by U.S. Patent 6,377,998 ("Noll"). Applicant respectfully contends for the following reasons that claims 47-54 and 56-62 are patentable over Noll.

Noll neither discloses nor suggests a network interface including a media access control ("MAC") that is programmable to perform a filtering operation on each packet of incoming frame data (to determine whether to accept the packet) and also programmable to perform at least one additional operation (that is not a filtering operation to determine whether to accept packets of incoming frame data) in response to incoming frame data, where the additional operation includes performance of a predetermined action in response to at least one predetermined bit pattern in the incoming frame data (as recited in claim 47) or includes transmission of diagnostic information (as recited in claim 50), or a method including a step of programming such a media access control to perform such an additional operation in response to incoming frame data (as recited in claim 56 or 59). Noll also fails to teach or suggest that such an "additional operation" in response to incoming frame data is transmission of packet data (as recited in claim 48 or 57) or issuance of a wake-up interrupt for a host computer (as recited in claim 49 or 58).

Noll discloses a frame processing apparatus (e.g., apparatus 200 of Fig. 2) including a MAC (e.g., MAC 208 of Fig. 2 or MAC 300 of Fig. 3A). Various elements of the MAC are programmable. For example, MAC 300 includes a programmable filter processor 312 (which can be programmed to perform any of various filtering operations on incoming frame data) and programmable protocol handlers 302. A stream of serial incoming frame data received by one of MAC 300's protocol handlers 302 undergoes serial-to-parallel conversion in receive state machine 360 (shown in Noll's Fig. 3B) and parallelized bytes of incoming frame data are asserted (via MAC 300's input multiplexer 306 and FIFO 310) to filter processor 312. Receive state machine 360 can also perform additional functions (described at Noll's column 11, lines 40-64, for example).

Applicant is unable to identify (and the Examiner has not identified) any teaching or suggestion in Noll that filter processor 312 or protocol handlers 302 (or any other element of MAC 300 or another MAC) should be programmed to perform (in response to incoming

frame data) an operation (that is not a filtering operation to determine whether to accept packets of the incoming frame data) including performance of a predetermined action in response to a predetermined bit pattern in the incoming frame data (as recited in claim 47 or 56) or transmission of diagnostic information (as recited in claim 50 or 59).

Noll teaches (e.g., at col. 3, lines 14-22) that the filter processor of its frame processing apparatus functions to filter data frames received by the frame processing apparatus "such that certain of the data frames are dropped and other data frames are provided with a switching destination." Such a function apparently corresponds to filtering operations (as recited) that determine whether to accept packets of incoming frame data. Noll teaches that filter processor 312 of Fig. 3A can be programmed to perform any of a variety of filtering operations to determine whether Noll's frame processing apparatus should accept frames of incoming frame data (e.g., store them in frame buffer 214 and forward them to some port of the apparatus) or not accept (i.e., drop) the frames. Noll apparently teaches (e.g., at col. 6, lines 13-55) that such filtering operations can be performed at the datalink layer to facilitate "level-2" switching (which apparently refers to forwarding of data received at one port to the same port or a different port implemented at "level-2" which is the "datalink" layer or level of functionality) or a layer higher than the datalink layer to facilitate "level-3" switching (e.g., including by extracting IP addresses from the incoming data and sending the IP addresses via bus 212 to forwarding tables 210).

Although Noll mentions (e.g., at col. 6, lines 13-55 and col. 6, line 60- col. 7, line 51) specific operations that can be performed by filter processor 312, these operations do not include any operation (performed in response to incoming frame data) that is not a filtering operation to determine whether to accept packets of incoming frame data but includes performance of a predetermined action in response to a predetermined bit pattern in incoming frame data (e.g., of any of the specific types recited in claims 48, 49, 57, and 58) or transmission of diagnostic information. Noll's teaching that a filter processor (e.g., filter processor 312) or MAC (e.g., MAC 208 or 300) can perform filtering operations on data being transmitted from a frame processing apparatus does not amount to a teaching or suggestion that the filter processor or MAC can or should perform any specific type of processing on incoming frame data being received by the frame processing apparatus.

With reference to claims 51-54 and 60-62, Noll fails to disclose or suggest a network interface comprising a MAC configured to receive incoming frame data from a network and a buffer manager coupled to receive the frame data from the MAC, wherein the MAC is operable in a first state in which it passes the incoming frame data to the buffer manager and in a second state in which it does not assert the incoming frame data to the buffer manager, and is configured to perform a filtering operation on a destination address of each packet of the incoming frame data before asserting all of the packet to the buffer manager and to assert all of the packet to the buffer manager only if the filtering operation results in a determination to accept the packet (as recited in claim 51). Nor does Noll teach or suggest a method for operating a network interface including a buffer manager and a MAC configured to receive incoming frame data from a network, wherein the MAC is operable in a first state in which it passes the incoming frame data to the buffer manager and is operable in a second state in which it does not assert the incoming frame data to the buffer manager, said method including the steps of: (a) operating the MAC to determine from a destination address of each packet of the incoming frame data whether to accept the packet, before asserting all of the packet to the buffer manager; and (b) asserting all of the packet from the MAC to the buffer manager only when step (a) results in a determination to accept the packet (as recited in claim 60).

The Office Action does not identify any teaching or suggestion by Noll of such limitations of claim 51 or 60, and Applicant is unable to identify any such teaching or suggestion in Noll. Noll's frame processing apparatus includes a frame buffer (e.g., frame buffer 214 of Fig. 2) but apparently does not include a "buffer manager" that is an element distinct from a MAC as recited in claims 51 and 60. Noll's teaching regarding the manner in which incoming frame data are asserted to frame buffer 214 is unclear. Noll's col. 5, lines 1-14, seem to teach (with reference to Fig. 2) that incoming frame data are asserted from MAC 208 via "data bus 216" to frame buffer 214, and that control signals are communicated via "control bus 222" between frame buffer 214 and switch circuitry 218. However, Noll's col. 22, lines 11-37, teach (with reference to Fig. 10) that frame buffer controller 1006 (of Fig. 10) within Noll's switch circuitry 218 (of Fig. 2) receives incoming frame data on data bus 216, and that frame buffer controller 1006 "stores and receives" incoming and outgoing data

frames "to the frame buffer 214 via the bus 222" shown in Fig. 10. It is not apparent whether and if so how the noted teaching of Noll's col. 5, lines 1-14, can be reconciled with that of Noll's col. 22, lines 11-37.

Although filter processor 312 of Noll's MAC 300 apparently performs a filtering operation on a destination address of each packet (frame) of incoming frame data, there is no teaching or suggestion determinable from Noll that filter processor 312 (or any other element of MAC 300 or 208) performs such a filtering operation before asserting all of the packet to a buffer manager (as recited in claims 51 and 60) or asserts all of the packet to such a buffer manager only if the filtering operation results in a determination to accept the packet (as recited in claims 51 and 60).

In view of the foregoing, Applicant respectfully requests consideration and allowance of claims 47-54 and 56-62 as amended.

Respectfully submitted,

GIRARD & EQUITZ LLP

Dated: 5/24/05 By: Alfred Equitz

Alfred A. Equitz
Reg. No.30,922

Attorneys for Applicant

ETHERNET TUTORIAL

ETHERNET FRAMES AND PACKETS

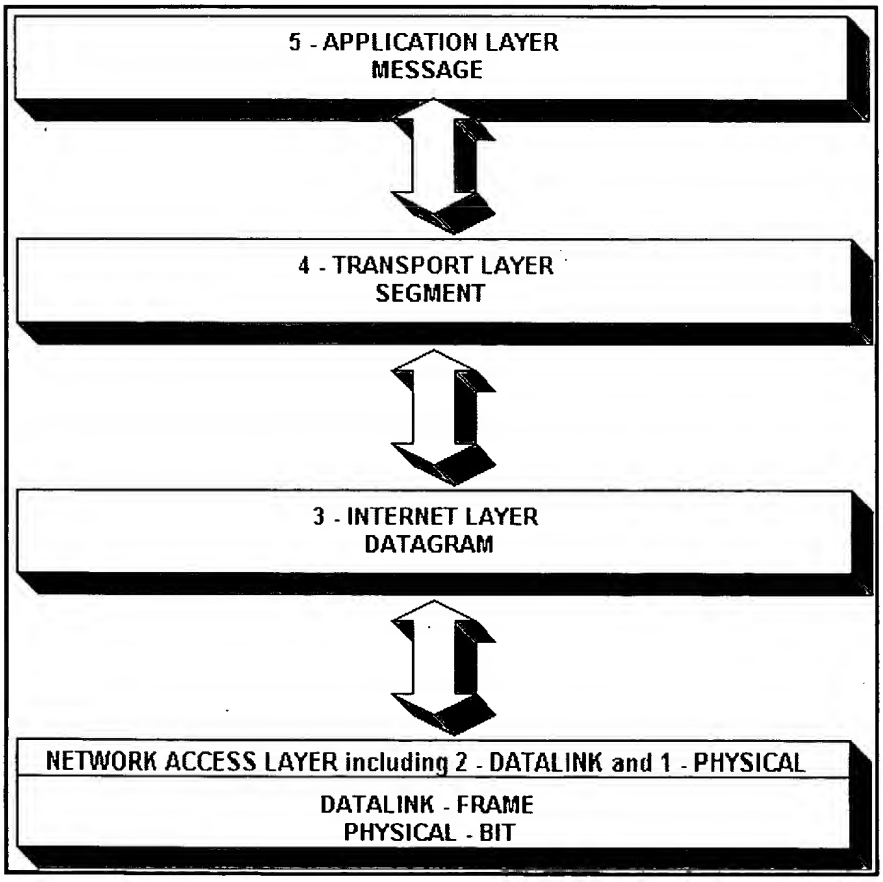
What are frames?

A frame can be defined as the **unit of data transferred across the network**, defined at the datalink (network access) layer of the protocol stack.

What are packets?

A packet can be defined as the **unit of data at any layer of the protocol stack**, prior to, or after transmission.

Is it really that simple?



It's not really quite that simple because some people prefer to further define data in terms of each layer of the TCP/IP protocol suite.

- Data at the **application layer** is called a **message**.
- **Transport layer** data is sometimes referred to as TCP or UDP **segments**.
- At the **internet layer**, data is sometimes talked of in terms of IP, ARP and RARP **datagrams**.

- The **datalink layer** data is referred to as **frames**.
- At the **physical layer**, data is sometimes referred to as **bits** - the lowest common denominator.

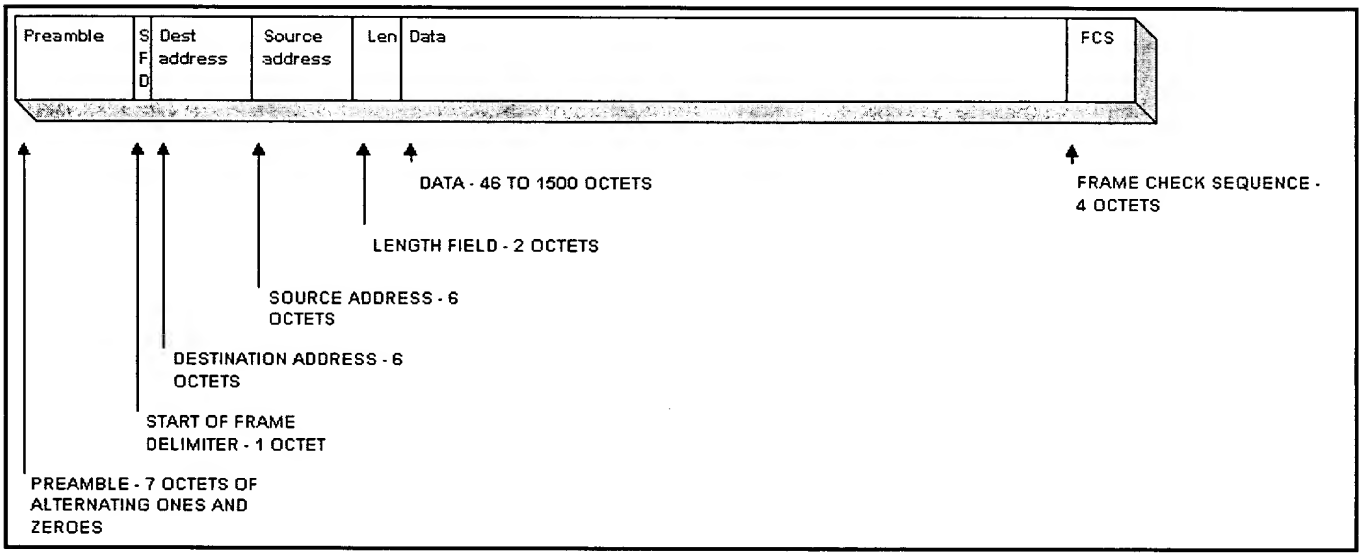
As data descends the source protocol stack, successive levels **encapsulate** the data with additional information. After a frame has been received at its destination, successive layers then **de-encapsulate** the data as it travels up the destination protocol stack.

Therefore, a packet can be discussed in terms of its subsets - messages, segments and datagrams.

Frames are not discussed in any other terms, they simply encapsulate the data with a header and addressing information.

What does a frame look like?

The diagram below illustrates the IEEE 802.3 frame:



An **octet** is defined as **eight bits** of data.

The ethernet Version 2.0 frame is similar to the IEEE 802.3 frame. For a device to be 802.3 compliant, it must be able to communicate with legacy Version 2.0 devices.

What does each field do?

- **Preamble** - alerts and synchronises the network interface card (NIC) to the incoming data.
- **Start of frame delimiter** - indicates the start of the frame.
- **Destination address** - the MAC (medium access control) address of the destination NIC. Three types of address exist:
  - **Unicast** - addresses a single device.
  - **Multicast** - addresses a group of devices.
  - **Broadcast** - addresses all devices.
- **Source address** - the MAC address of the source NIC.
- **Length** - indicates the length of the data field.

- **Data** - carries the data being transferred. It has a maximum limit to stop devices sending too much data at any time. This gives fair access to the media to all devices.

- **Frame check sequence** - provides a cyclic redundancy check (CRC) on all data held in the frame. CRC is an error detection mechanism generated by the NICs. The source NIC generates a 32 bit CRC figure from the address, type and data fields. The destination NIC does the same calculations. If the destination NIC calculates the same figure for the CRC as the source, the frame was received error-free.

Tell me more!

Ethernet and IEEE 802.3 Frame Formats	From the Cisco web site
Kansas State University	Basic frame details
Temple University frame FAQ	Easy to follow slide show

HOME BACK CONTENTS SITEMAP